An implicit filtering algorithm for derivative-free multiobjective optimization with box constraints

An implicit filtering algorithm for derivative-free multiobjective optimization with box constraints

G. Cocchi*, G. Liuzzi†, A. Papini‡, M. Sciandrone*

November 16, 2017

Abstract. This paper is concerned with the definition of new derivative-free methods for box constrained multiobjective optimization. The method that we propose is a non-trivial extension of the well-known implicit filtering algorithm to the multiobjective case. Global convergence results are stated under smooth assumptions on the objective functions. We also show how the proposed method can be used as a tool to enhance the performance of the Direct MultiSearch (DMS) algorithm. Numerical results on a set of test problems show the efficiency of the implicit filtering algorithm when used to find a single Pareto solution of the problem. Furthermore, we also show through numerical experience that the proposed algorithm improves the performance of DMS alone when used to reconstruct the entire Pareto front.

Keywords: Multiobjective nonlinear programming, derivative-free optimization, implicit filtering

AMS subject classification: 90C30, 90C56, 65K05

1 Introduction

Many engineering and financial problems do not fit well in the classical optimization framework where the minimization (or maximization) of a single objective function, subject to some constraints on the variables, is pursued. Indeed, many applications require the optimization of two (or even more), often conflicting, objective functions at the same time. Such problems are known in the literature as multiobjective optimization (MO) problems (see, e.g., [18]). One remarkable peculiarity of MO problems, with respect to single objective ones, is that it is possible to conceptually distinguish between the problem solver and the decision maker. This is because, as we will see, a MO problem can have (and indeed it has) many "equivalent" (so-called non-dominated) solutions. More precisely, such solutions are equivalent for what concerns the problem solver (they are kind of indistinguishable one from another) but, they can be very different with respect to the decision maker. Such distinction of roles between the problem solver and the decision maker is so profound in the context of MO that methods for the solution of MO problems can be roughly classified with respect to the moment when preferences of the decision maker are established. In this respect, we have:

- *methods without preferences*, in which the preferences of the decision maker are completely disregarded and finding any multiobjective solution is considered acceptable;

*Dipartimento di Ingegneria dell'Informazione, Università di Firenze, Via di Santa Marta 3, 50139 Firenze (Italy)
†Istituto di Analisi dei Sistemi ed Informatica, Consiglio Nazionale delle Ricerche, Via dei Taurini 19, 00185 Roma (Italy)
‡Dipartimento di Ingegneria Industriale, Università di Firenze, Viale Morgagni 40/44, 50134 Firenze (Italy)

- *methods with a-priori statement of preferences*, in which decision maker preferences are needed in order to reduce the solution of a multiobjective problem to the solution of (a sequence of) single objective problems, through a process called scalarization; the process thus provide a single multiobjective solution, which depends on the parameters chosen by the decision maker to combine the objective functions into a scalar one.

- *methods with a-posteriori employment of preferences*, in which decision maker preferences are considered at the end of the optimization process in order to choose the best solution out of a set of equivalent (non-dominated) solutions; the richer the set of final solutions presented to the decision maker, the better, i.e. the more freedom the decision maker has when selecting her/his preferred solution.

In this paper, we consider the following bound-constrained MO problem

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} (f_1(x), f_2(x), \ldots, f_m(x)) \\ &s.t.\ l_i \le x_i \le u_i, \quad i = 1, \ldots, n, \end{aligned} \tag{1}$$

where $f_i : \mathbb{R}^n \mapsto \mathbb{R}$, $i = 1, \ldots, m$, are $m \geq 2$ real-valued objective functions, and $l_i, u_i \in \mathbb{R}$ with $l_i < u_i$.

In particular, even though we require each objective function to be continuously differentiable, we assume them to be of the *black-box type*, i.e. first (and higher) order derivatives cannot be directly computed nor approximated in any way. We remark that this situation is common in many applications, see e.g. [3]. Indeed, derivatives of the functions defining the problem could be impossible or too costly to obtain, for instance when objective functions values are computed by time-consuming and complex simulation programs. Furthermore, it should be noted that, even in those cases where finite differences are affordable and could ideally be used, approximated derivatives could be untrustworthy and useless simply because the problem functions are affected by noise. In such cases, the so-called derivative-free methods can be used to solve the problem. Hence, we are interested in the development and analysis of *derivative-free methods* for MO problems.

When a single objective is to be minimized, one of the most popular and efficient derivative-free methods is the so-called implicit filtering algorithm, originally proposed in [12] and applied to the solution of a variety of optimal design problems, see e.g. [1, 2, 5, 7, 9, 10, 13]. The implicit filtering algorithm is basically a finite-difference gradient-based method in that it makes use of gradient approximations obtained by finite differences. However, what makes the algorithm well-suited for the class of problems we are interested in is that the step size in the difference is adaptively chosen by the algorithm itself. More specifically, at the very beginning of the optimization process the differentiation step size h is chosen to be relatively large. Then, when convergence of the method is deemed with a given step size, the step is reduced so that, by computing new and more accurate gradient approximations, progress of the iterates toward a solution can resume again.

When more than one objective is present and derivatives cannot be used, a wide variety of methods can be employed. Stochastic or probabilistic methods, like e.g., genetic or evolutive algorithms (see e.g., [11, 20] and the references therein), have long been proposed and are available. More recently, in [4, 16], deterministic algorithms for problem (1) have been proposed which export to the multiobjective context derivative-free techniques for single objective problems. In particular, in the relatively recent paper [4], a direct multisearch (DMS) algorithm has been proposed. DMS extends to the multiobjective case the well-known direct search (or pattern search) paradigm. In [4], it has been shown that DMS has strong theoretical convergence properties and that, most importantly, such stronger properties reflect in better numerical performances of DMS with respect to a large selection of the most used and well-known stochastic algorithms.

Our aim in the paper is therefore two-fold. On the one hand, even though the implicit filtering algorithm cannot be directly used to solve problem (1), we approach the problem as in [8], where a steepest-descent method for problem (1) is proposed, but by approximating objective derivatives as

in the implicit filtering strategy. Hence, we define a new method *without preferences* for the solution of problem (1). The method exports the derivative-free skills of the implicit filtering approach [12, 14] within the steepest descent framework proposed in [8] for multiobjective optimization. For this method, we also prove convergence to a "stationary" point of problem (1). On the other hand, we also propose a version of the algorithm which is suitable to generate a set of non-dominated solutions, thus approximating the set of Pareto solutions of problem (1), rather than a single point. The paper is organized as follows. In section 2, we report some preliminaries about multiobjective optimization. In section 3, we recall the implicit filtering algorithm for (single objective) derivative-free optimization. In section 4, we define the multiobjective implicit filtering algorithm. Global convergence results are stated in section 5. In section 6, we perform a computational experimentation of the proposed method and a comparison with the DMS algorithm proposed in [4]. In section 6.2, we show how the multiobjective implicit filtering algorithm can be used within DMS to improve its ability to generate the Pareto front of the problem. Finally, section 7 contains some concluding remarks. In the Appendix we give the proofs of two results used for the convergence analysis.
We conclude this section by recalling some notation that will be useful later. With reference to problem (1), we denote by $F : \mathbb{R}^n \to \mathbb{R}^m$ the vector-valued function defined by

$$F(x) \stackrel{\triangle}{=} (f_1(x), f_2(x), \ldots, f_m(x))^\top,$$

and by $J : \mathbb{R}^n \to \mathbb{R}^{m\times n}$ its Jacobian matrix function,

$$J(x) = (\nabla f_1(x), \nabla f_2(x), \ldots, \nabla f_m(x))^\top.$$

Furthermore, we denote by $\mathcal{F}$ the feasible set of Problem (1), i.e.,

$$\mathcal{F} = \{x \in \mathbb{R}^n : l_i \le x_i \le u_i,\ i = 1, \ldots, n\} \tag{2}$$

and note that $\mathcal{F}$ is, by definition, compact.
Given any two vectors $u, v \in \mathbb{R}^p$,

$$\begin{aligned} u < v &\quad\Leftrightarrow\quad u_i < v_i, \text{ for all } i = 1, \ldots, p \\ u \leqq v &\quad\Leftrightarrow\quad u_i \le v_i, \text{ for all } i = 1, \ldots, p \\ u \le v &\quad\Leftrightarrow\quad u \leqq v \text{ and } u \ne v. \end{aligned}$$

Finally, let us denote by $e_i \in \mathbb{R}^n$, $i = 1, \ldots, n$, the vectors that form the canonical basis in $\mathbb{R}^n$, and by $\mathbf{1}$ the vector, of appropriate dimension, of all ones, e.g., in $\mathbb{R}^n$, $\mathbf{1} = \sum_{i=1}^n e_i$.

2 Preliminaries on multiobjective optimization

Just as in single objective optimization, in the context of multiobjective optimization we should be able to compare two vectors $x, y \in \mathbb{R}^n$ on the basis of their respective vectors of function values $F(x), F(y)$. To this aim, the following definition of (Pareto) dominance is useful.

Definition 1 (Pareto dominance) *Given two vectors $x, y \in \mathbb{R}^n$, we say that x (strictly) Pareto dominates y when*

$$F(x) \leqq F(y) \ \big(F(x) \le F(y)\big).$$

Then, with reference to problem (1), an ideal solution would be a point $x^* \in \mathcal{F}$ such that x^* Pareto dominates each other feasible point $x \in \mathcal{F}$, i.e.,

$$F(x^*) \leqq F(x), \qquad \text{for all } x \in \mathcal{F}.$$

Unfortunately, such a point x^* very seldom exists, which is why the following definitions of optimality are introduced for multiobjective problems.

Definition 2 (Weak Pareto optimality) *A point $x^* \in \mathcal{F}$ is a weak Pareto optimal point for Problem (1), if there does not exist any $x \in \mathcal{F}$ such that*

$$F(x) < F(x^*).$$

Definition 3 (Pareto optimality) *A point $x^* \in \mathcal{F}$ is a Pareto optimal point for Problem (1), if there does not exist any $x \in \mathcal{F}$ such that*

$$F(x) \leq F(x^*).$$

By means of these two definitions, we are able to identify a set of non-dominated points (the so-called Pareto front or frontier) which is constituted by the "optimal" solutions of the multiobjective problem (1).
Just as in the single-objective case, to define solution algorithms and analyze their convergence properties, we need to introduce the definition of *Pareto-stationarity.*

Definition 4 (Pareto stationarity) *A point $x^* \in \mathcal{F}$ is Pareto stationary for Problem (1) if, for all $y \in \mathcal{F}$, an index $j \in \{1, \ldots, m\}$ exists such that*

$$\nabla f_j(x^*)^T (y - x^*) \geq 0.$$

It can be easily shown that, if x^* is a Pareto optimal point for Problem (1), then x^* is Pareto stationary for Problem (1) (the inverse implication can only be shown when F is a convex continuously differentiable map, as in the single objective case).

Thanks to Definition 4, when $\bar{x} \in \mathcal{F}$ is not a Pareto stationary point, we know that an $y \in \mathcal{F}$ must exist such that $v = y - \bar{x}$ is a descent direction for all the objective functions f_i, $i = 1, \ldots, m$, at $\bar{x}$.
For any given $x \in \mathcal{F}$, we define the function $g_x : \mathcal{F} \to \mathbb{R}^m$ by

$$g_x(y) = \max_{i=1,\ldots,m} \nabla f_i(x)^\top (y - x),$$

and note that g_x is continuous, piecewise linear, and convex. By compactness of $\mathcal{F}$, g_x admits a global minimum on $\mathcal{F}$; hence, we denote respectively by $\theta(x)$ and $y(x)$ the global minimum value and a global minimum point of g_x over $\mathcal{F}$, i.e.,

$$\theta(x) = \min_{y \in \mathcal{F}} g_x(y) \tag{3}$$

$$y(x) = \arg\min_{y \in \mathcal{F}} g_x(y). \tag{4}$$

For what concerns the above definitions, we remark that problem (3) is a finite *minimax* problem with linear component functions. It can thus be trivially restated as the following linear programming problem

$$\begin{aligned} \min_{y,\beta} \ & \beta \\ & \nabla f_i(x)^\top (y - x) \leq \beta, \quad i = 1, \ldots, m, \\ & y \in \mathcal{F} \end{aligned} \tag{5}$$

Furthermore, from [8], we report the following proposition.

Proposition 1 *Given problem (1), let $\theta : \mathcal{F} \to \mathbb{R}$ be defined as in (3). Then the following statements hold:*

- *θ is a continuous function;*
- *$\theta(x) \leq 0$, for all $x \in \mathcal{F}$;*

- $x^* \in \mathcal{F}$ *is Pareto stationary for Problem (1) if and only if* $\theta(x^*) = 0$.

We conclude this section by formally introducing the steepest descent direction for the vector valued mapping F at x.

Definition 5 (Steepest descent direction) *Given any point* $x \in \mathcal{F}$*, the steepest descent direction for* F *at* x *is*

$$v(x) = y(x) - x$$

where $y(x)$ *is given by (4).*

On the basis of Definition 5, from reference [8], we recall the following steepest descent algorithm for the solution of Problem (1) (we recall from the introduction that $J(x)$ denotes the Jacobian of the vector of objective functions).

Algorithm 1 `Steepest_Descent`

Data: $x_0 \in \mathbb{R}^n$, $\gamma \in (0, 1)$
for $k = 0, 1, \ldots,$ **do**
  Compute $\theta(x_k)$ and $v(x_k)$
  if $\theta(x_k) = 0$ **then**
    x_k is Pareto stationary, STOP
  end if
  Compute $\alpha_k = 2^{-\beta_k}$ with β_k the smallest non-negative integer s.t.

$$F(x_k + \alpha_k v(x_k)) \leq F(x_k) + \gamma \alpha_k J(x_k) v(x_k).$$

  Set $x_{k+1} = x_k + \alpha_k v(x_k)$
end for

For a convergence analysis of Algorithm 1, we refer the reader to [8].

3 The implicit filtering algorithm for unconstrained single objective optimization

In this section we briefly recall the main concepts of the implicit filtering algorithm for derivative-free optimization (see [14], and the references therein, for a more thorough description of the algorithm). To this aim and limited to this section, let us consider the unconstrained (single-objective) problem

$$\min_{x \in \mathbb{R}^n} \; f(x) \tag{6}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function. In order to introduce the implicit filtering algorithm for the solution of Problem (6), which is basically an improvement of the well-known coordinate search algorithm, we need to recall some definitions.

Definition 6 (Stencil) *Given a point* x *and a stepsize* $h > 0$*, a* stencil $S(x, h)$ *is a set of points, namely*

$$S(x, h) = \{x + he_1, \ldots, x + he_n, x - he_1, \ldots, x - he_n\}.$$

Then, with reference to f, an approximation of $\nabla f(x)$ can be obtained by using the information obtained by evaluating f onto the points belonging to the stencil $S(x, h)$. More precisely, let us introduce the definition of approximated gradient of f at x.

Definition 7 (Approximated gradient) *Given x, $h > 0$ and the stencil $S(x,h)$, the approximated gradient $\nabla_h f$ of f at x is defined by*

$$\nabla_h f(x) \triangleq \left(\frac{\partial_h f(x)}{\partial x_1}, \ldots, \frac{\partial_h f(x)}{\partial x_n} \right)^\top$$

where, for all $i = 1, \ldots, n$,

$$\frac{\partial_h f(x)}{\partial x_i} = \frac{f(x + he_i) - f(x - he_i)}{2h}.$$

It is worth noting that, since f is continuously differentiable, $\nabla_h f$ is continuous for every $h \geq 0$. Then, let us recall the definition of a "stencil failure".

Definition 8 (Stencil failure) *Given x and a stepsize $h > 0$, a stencil-failure at x occurs when*

$$f(x) \leq f(y), \qquad \forall y \in S(x,h), \tag{7}$$

i.e. no point y in the stencil $S(x,h)$ improves the objective function with respect to the value $f(x)$.

Now, we are ready to introduce the sketch of a particular instance of the well-known implicit filtering algorithm (see [14] for more general algorithms), which basically consists of two nested loops. The external loop (see Algorithm 2) produces a sequence of stepsizes $\{h_k\}$, such that $h_k \to 0$ for $k \to \infty$, and a sequence of iterates $\{x_k\}$; each x_k is obtained by performing a single implicit filtering step, `imstep`, with fixed stepsize h_k. The inner loop (see Algorithm 3) performs a minimization of the objective function f starting from the current iterate $x_k = z_0$. At every iteration, first a stencil is built around z_j. If a stencil failure is not detected, i.e. at least one of the stencil points $y \in S(z_j, h_k)$ strictly decreases the objective function, the approximated gradient is computed. Then, a linesearch is carried out to obtain a new iterate z_{j+1}, provided that the current point is not h_k-stationary. We note that, in Algorithm 3, when the linesearch is performed, the "approximated" steepest descent direction is used. In case of failure of the linesearch, z_{j+1} is set equal to y. The algorithm keeps iterating through the inner cycle until either a stencil failure occurs, or the current point z_j is deemed h_k-stationary. For a thorough convergence analysis of Algorithm 2, as well as for less basic versions of the implicit filtering method, we refer the interested reader to [14].

Algorithm 2 `Implicit_Filtering`

Data: $x_0 \in \mathbb{R}^n$, $h_0 > 0$, $\gamma, \delta, \tau \in (0,1)$
for $k = 0, 1, \ldots,$ **do**
  Set $x_{k+1} =$`imstep`(x_k, h_k, τ, γ)
  Set $h_{k+1} = \delta h_k$
end for

4 Implicit filtering for multiobjective optimization

Our main goal in the paper is to define an implicit filtering-type method to *solve* the multiobjective optimization problem (1). Indeed, we are able to prove that the proposed method converges to Pareto stationary points of (1). In order to export the implicit filtering algorithm to a multiobjective context, at least four critical aspects must be carefully considered which are:

- a new definition of a stencil failure;
- handling of "quasi" (Pareto) stationarity;

Algorithm 3 `imstep`(z_0, h, τ, γ)

for $j = 0, 1, \ldots$ **do**
  Build $S(z_j, h)$ and define $\mathcal{S} = \{y \in S(z_j, h) : f(y) < f(z_j)\}$
  if $\mathcal{S} \neq \emptyset$ **then**
    Choose $y \in \mathcal{S}$
  else
    return z_j (*stencil failure: reduce the stepsize*)
  end if
  Compute $\nabla_h f(z_j)$
  if $||\nabla_h f(z_j)|| \leq \tau h$ **then**
    return z_j (*h-stationarity: reduce the stepsize*)
  else if $\alpha > 0$ exists such that $f(z_j - \alpha \nabla_h f(z_j)) \leq f(z_j) - \gamma\alpha||\nabla_h f(z_j)||^2$ **then**
    Set $z_{j+1} = z_j - \alpha \nabla_h f(z_j)$
  else
    Set $z_{j+1} = y$
  end if
end for

- approximation of the multiobjective steepest descent direction;
- a multiobjective linesearch.

Moreover, bound constraints on the variables must be taken into account in the definition of stencil and approximated gradients.

Definition 9 (Approximated gradient) *Given $x \in \mathcal{F}$, $h > 0$, the stencil $S(x, h)$, and an index $i \in \{1, \ldots, m\}$, the approximated gradient $\nabla_h f_i$ of f_i at x is defined by*

$$\nabla_h f_i(x) \triangleq \left(\frac{\partial_h f_i(x)}{\partial x_1}, \ldots, \frac{\partial_h f_i(x)}{\partial x_n} \right)^\top$$

where, for all $j = 1, \ldots, n$,

$$\frac{\partial_h f_i(x)}{\partial x_j} = \begin{cases} \frac{f_i(x+he_j) - f_i(x-he_j)}{2h}, & x + he_j \in \mathcal{F},\ x - he_j \in \mathcal{F}; \\ \frac{f_i(x+he_j) - f_i(x)}{h}, & x + he_j \in \mathcal{F},\ x - he_j \notin \mathcal{F}; \\ \frac{f_i(x) - f_i(x-he_j)}{h}, & x + he_j \notin \mathcal{F},\ x - he_j \in \mathcal{F}; \\ \infty, & x + he_j \notin \mathcal{F},\ x - he_j \notin \mathcal{F}. \end{cases} \tag{8}$$

Note that, $\frac{\partial_h f_i(x)}{\partial x_j} = \infty$ whenever $x + he_j \notin \mathcal{F},\ x - he_j \notin \mathcal{F}$, but this never occurs for sufficiently small values of h; we clarify this point in Lemma 4 below.
Now, we introduce the definition of a *stencil failure* in the multiobjective case and of Pareto h-stationarity.

Definition 10 (Multiobjective stencil failure) *Given $x \in \mathcal{F}$ and a stepsize $h > 0$, a stencil failure at x occurs when there is no $y \in S(x, h) \cap \mathcal{F}$ that dominates x, i.e.*

$$\nexists y \in S(x, h) \cap \mathcal{F} : \quad F(y) \quad \leq \quad F(x).$$

Definition 11 (Pareto h-stationarity) *Given a stepsize $h > 0$, a point $x^* \in \mathcal{F}$ is Pareto h-stationary if, for all $y \in \mathcal{F}$, an index $j \in \{1, \ldots, m\}$ exists, possibly depending on y, such that*

$$\nabla_h f_j(x^*)^T (y - x^*) \geq 0.$$

Reasoning as in Section 2, Pareto h-stationarity can be characterized by extending definitions (3-4) for $\theta(x)$ and $y(x)$ to the case of approximated gradients, as follows:

$$\theta(x,h) = \min_{y\in\mathcal{F}} \max_{i=1,\dots,m} \nabla_h f_i(x)^\top (y-x) \tag{9}$$

$$y(x,h) = \arg\min_{y\in\mathcal{F}} \max_{i=1,\dots,m} \nabla_h f_i(x)^\top (y-x). \tag{10}$$

Further, we approximate the steepest descent direction $v(x)$ by

$$v(x,h) = y(x,h) - x.$$

It can be easily seen (see point (i) of Proposition 3 in the Appendix) that $\theta(x,h) \leq 0$ for all $x \in \mathcal{F}$ and $h > 0$, so that x^* is Pareto h-stationary if and only if $\theta(x^*,h) = 0$. Furthermore, (see point (ii) of Proposition 3 in the Appendix) for any given $x \in \mathcal{F}$, it results

$$\lim_{h\to 0^+} \theta(x,h) = \theta(x).$$

Now, we develop a first version of an implicit filtering algorithm which converges asymptotically to a single Pareto stationary point. As in the implicit filtering framework, our algorithm consists of two nested loops. The outermost loop, which is reported in Algorithm 4, is very similar to the outer loop of Algorithm 2 for single objective optimization reported in Section 3.

Algorithm 4 `MultiObjectiveImplicitFiltering (MOIF)`

Data: $x_0 \in \mathcal{F}$, $h_0 > 0$, $\gamma, \delta, \tau \in (0,1)$
for $k = 0, 1, 2, \dots$ **do**
  $x_{k+1} =$ `imstepMulti`(x_k, h_k, τ, γ)
  $h_{k+1} = \delta h_k$
end for

The innermost loop is the most important one and is reported in the following algorithm `imstepMulti`. As we can see, the internal loop is somewhat different from Algorithm 3 and is composed by two parts:

- direct search along the coordinate axis;
- (possible) linesearch along the approximated steepest descent direction (obtained by solving an LP).

The main difference between Algorithms 3 and 5 resides in the behavior of the algorithms in case of a stencil failure. When a stencil failure is detected by Algorithm 3, the inner loop is abandoned so that the step size is reduced. On the contrary, in Algorithm 5, when a stencil failure is detected, the inner loop tries to execute a linesearch along the approximated steepest descent direction.
We decided to modify the basic implicit filtering behavior by drawing inspiration from the recently proposed algorithm DMS (see Remark 4 below). We move along coordinate axis by a fixed step h. At iteration j of Algorithm `imstepMulti`, we have a stencil failure when no feasible stencil point $y \in S(w,h) \cap \mathcal{F}$ exists such that

$$F(y) \leq F(w) - \mathbf{1}\gamma h.$$

When a stencil failure finally occurs, we proceed trying to find a descent direction using $\nabla_h F(z_j)$. In particular, θ_j and $v_j = y_j - z_j$ are computed by solving a linear programming subproblem of type (5) with exact gradients replaced by approximated ones. In this respect, we note that this calculation does not increase the number of objective function evaluations, because the approximated gradient is computed through stencil points already computed.
If we find a sufficiently good feasible direction v_j, i.e. such that $\theta(z_j,h) < -\tau h$, we perform a linesearch along the computed direction v_j. We recall that direction v_j is such that the unitary stepsize is feasible, i.e. $z_j + v_j \in \mathcal{F}$ and v_j is a feasible direction.

Algorithm 5 `imstepMulti`$(\tilde{z}_0, h, \tau, \gamma)$

for $j = 0, 1, \ldots$ **do**
  Set `stencilFailure` = FALSE and $w = \tilde{z}_j$
  while (**not** `stencilFailure`) **do**
    Build $S(w, h)$
    if $S(w, h) \cap \mathcal{F} = \emptyset$ **then**
      return w (*the stencil is infeasible: reduce the stepsize*)
    else
      define $\mathcal{S} = \{y \in S(w, h) \cap \mathcal{F} : F(y) \leq F(w) - \mathbf{1}\gamma h\}$
    end if
    if $\mathcal{S} \neq \emptyset$ **then**
      Choose $y \in \mathcal{S}$ and set $w = y$
    else
      Set `stencilFailure`=TRUE
    end if
  end while
  Set $z_j = w$
  if $\exists i \in \{1, \ldots, n\}$ s.t. $z_j + he_i, z_j - he_i \notin \mathcal{F}$ **then**
    return z_j (*approximated gradient undetermined: reduce the stepsize*)
  end if
  Compute $\theta_j = \theta(z_j, h)$, $y_j = y(z_j, h)$, $v_j = y_j - z_j$
  N.B. at this point, either *i)* $z_j = \tilde{z}_j$ or *ii)* $z_j \neq \tilde{z}_j$ and $F(z_j) \leq F(\tilde{z}_j) - \mathbf{1}\gamma h$.
  if $\theta_j \geq -\tau h$ **then**
    return z_j (*Pareto h-stationarity: reduce the stepsize*)
  end if
  Compute $\alpha_j = \text{Goldstein}(z_j, v_j, h, \theta_j, \gamma)$
  if $\alpha_j |\theta_j| \leq \tau h$ **then**
    return z_j (*line search-failure: reduce the stepsize*)
  end if
  Set $\tilde{z}_{j+1} = z_j + \alpha_j v_j$
end for

Remark 1 *If $\theta(z_j, h) \geq -\tau h$, the current point z_j is deemed h-stationary and the current stepsize h is reduced by the algorithm. This could also mean that the current gradient approximation is not suitable. Further, even if $\theta(z_j, h) < -\tau h < 0$, there is no guarantee that direction v_j is a descent direction for the objective functions.*

Remark 2 *Direction v_j is computed by using approximations of the objective functions gradients. Hence, for a fixed stepsize h, it might well not be a descent direction at the current iterate z_j.*

The above two observations suggested us to use a Goldstein linesearch (in place of an Armijo one) with initial stepsize α_j equal to h, i.e. the stepsize used in the algorithm to build the stencil and compute the approximated objective functions gradients. In this way, we are able to immediately verify the quality of the direction, thus avoiding a potentially large number of function evaluations.

Remark 3 *We observe that, whenever the test $\alpha_j |\theta_j| \leq \tau h$ holds, as $|\theta_j| > \tau h$, we have that the produced α_j is "sufficiently small", so that, the current point can be considered an approximation of a Pareto h-stationary point and, as a consequence, the finite-difference stepsize h is reduced.*

Remark 4 *As we said before, the direct search along the coordinate directions draws inspiration from DMS, which is an algorithmic framework that extends the class of direct search methods for single optimization to nonsmooth, multiobjective optimization. In the general setting of DMS, the*

Algorithm 6 `Goldstein`$(x, v, h, \theta, \gamma)$

if $F(x + hv) \leq F(x) + \mathbf{1}\gamma(h\theta)$ **and** $x + hv \in \mathcal{F}$ **then**
  Set $\beta \leftarrow 0$
  while $x + 2^{\beta+1}hv \in \mathcal{F}$ and $F(x + 2^{\beta+1}hv) \leq F(x) + \mathbf{1}\gamma(2^{\beta+1}h\theta)$ **do**
    Set $\beta \leftarrow \beta + 1$
  end while
  return $\alpha = 2^{\beta}h$
else
  return $\alpha = 0$
end if

objective functions are sampled along suitable sets of search directions, the acceptance criterion is based on the Pareto dominance, and a list of feasible nondominated points is updated. The simplest strategy of the framework is that of considering lists formed by a single point and to use the set of coordinate directions as search directions. This strategy is similar to that defined in the direct search block of our algorithm. However, in our framework the implicit filtering phase generates an approximated "steepest descent direction" for all objective functions, by solving the min max *problem (see (10)), and this is the key ingredient, coupled with the line search, to ensure convergence properties.*

5 Convergence analysis

First of all we prove that Algorithm `imstepMulti` is well defined. To this aim, in the following lemma we show that for every $j \geq 0$ and $\tilde{z}_j$ a point z_j is produced, i.e. `stencilFailure` is eventually set to $TRUE$.

Lemma 1 *With reference to Algorithm* `imstepMulti`*, for every $j \geq 0$, after a finite number of iterations of the while-loop, either stencil infeasibility is detected or stencil failure is obtained, i.e.* `stencilFailure` *is eventually set to $TRUE$.*

Proof. Let us assume by contradiction that for a given iteration j the while-loop never terminates, i.e. the stencil is always feasible and `stencilFailure` is never set to TRUE. In this case, let us denote by w_t the stencil center point at the generic t-th iteration of the while loop, and by w_{t+1} the point of the next stencil. Of course, for every t it results

i) $w_t \in \mathcal{F}$;

ii) $F(w_{t+1}) \leq F(w_t) - \mathbf{1}\gamma h$.

Recalling that $w_0 = \tilde{z}_j$ and point ii) above, we can write

$$F(w_{t+1}) \leq F(w_t) - \mathbf{1}\gamma h \leq F(w_{t-1}) - \mathbf{1}(2\gamma h) \leq \cdots \leq F(\tilde{z}_j) - \mathbf{1}(t+1)\gamma h.$$

Taking the limit for $t \to \infty$ in the above relation we would obtain

$$\lim_{t\to\infty} f_i(w_t) = -\infty, \quad i = 1, \ldots, m,$$

which is a contradiction with the continuity of F, compactness of $\mathcal{F}$, and $\{w_t\} \subset \mathcal{F}$. □

By virtue of Lemma 1,

- either $z_j = \tilde{z}_j$ and $F(z_j) = F(\tilde{z}_j)$
- or z_j is such that $F(z_j) \leq F(\tilde{z}_j) - \mathbf{1}\gamma h$.

Then, we prove that the Goldstein procedure is well-defined, namely that the while-loop in the Goldstein procedure cannot infinitely cycle.

Lemma 2 *For every $x \in \mathcal{F}$, $h > 0$, $\theta < 0$, and $v \in \mathbb{R}^n$, the Goldstein procedure always returns a value $\alpha \geq 0$.*

Proof. If $F(x + hv) \not\leq F(x) + \mathbf{1}\gamma(h\theta)$ or $x + hv \notin \mathcal{F}$, the procedure immediately returns $\alpha = 0$. Hence, we have only to analyze the case when $F(x + hv) \leq F(x) + \mathbf{1}\gamma(h\theta)$ and $x + hv \in \mathcal{F}$, and show that the while-loop cannot infinitely cycle. Let us proceed by contradiction and assume that the while-loop infinitely cycle. This means that, for every $\beta = 0, 1, 2, \ldots$, we have

$$x + 2^{\beta+1}hv \in \mathcal{F} \text{ and } F(x + 2^{\beta+1}hv) \leq F(x) + \mathbf{1}\gamma(2^{\beta+1}h\theta).$$

For β sufficiently large, this is a contradiction with the compactness of $\mathcal{F}$ and continuity of F, and concludes the proof. □

Now, concerning the convergence of Algorithm `MOIF` (the `Multi Objective Implicit Filtering` procedure described in Algorithm 4), the first thing we need to prove is that the stepsize h_k converges to zero. Then, we state the following lemma.

Lemma 3 *Algorithm `MOIF` generates infinite sequences $\{x_k\} \subset \mathcal{F}$ and $\{h_k\} \subset \mathbb{R}^+$, such that*

$$\lim_{k\to\infty} h_k = 0. \tag{11}$$

Proof. We assume by contradiction that for an index $\bar{k}$ Algorithm `imstepMulti` does not terminate, thus producing infinite sequences $\{\tilde{z}_j\}$, $\{z_j\}$, $\{\alpha_j\}$, $\{\theta_j\}$, $\{v_j\}$, and $\{f_i(z_j)\}$, for $i = 1, \ldots, m$. Let $\bar{h} = h_{\bar{k}}$. Furthermore, for all $j = 0, 1, \ldots,$

i) $\theta_j = \theta(z_j, \bar{h}) < -\tau\bar{h} < 0$, and

ii) $z_j + \alpha_j v_j \in \mathcal{F}$ and $\alpha_j|\theta_j| > \tau\bar{h}$.

Hence, for all $i \in \{1, \ldots, m\}$ and $j \geq 0$, we have

$$f_i(z_{j+1}) \leq f_i(\tilde{z}_{j+1}) = f_i(z_j + \alpha_j v_j) \leq f_i(z_j) + \gamma\alpha_j\theta_j,$$

that is

$$f_i(z_j) - f_i(z_{j+1}) \geq -\gamma\alpha_j\theta_j. \tag{12}$$

Since $\{f_i(z_j)\}$, for $i = 1, \ldots, m$, are non-increasing sequences, and $z_j \in \mathcal{F}$ for all j, then the continuity of f_i ensures that

$$\lim_{j\to\infty} f_i(z_j) - f_i(z_{j+1}) = 0, \ \ \forall i = 1, \ldots, m.$$

Above limits and relation (12) imply that

$$\lim_{j\to\infty} \alpha_j|\theta_j| = 0.$$

Thus, for j sufficiently large we will have $\alpha_j|\theta_j| \leq \tau\bar{h}$. In other words, the test after the Goldstein procedure will be satisfied, and hence Algorithm `imstepMulti` will terminate, within a finite number of inner iterations, in contradiction with our initial assumption. Then (11) is easily proved, as $\{h_k\}$ is an infinite sequence such that $h_{k+1} = \delta h_k$ with $\delta \in (0, 1)$. □

Before stating the main theorem, we prove the following result.

Lemma 4 *Let $\{x_k\}_K$ be a subsequence produced by Algorithm* `MOIF` *and assume that $x_k \to \bar{x}$ for $k \in K$ and $k \to \infty$. Then, for $k \in K$ and k sufficiently large we have that for $j = 1, \ldots, n$ at least one of the following conditions holds*

$$x_k + h_k e_j \in \mathcal{F}$$
$$x_k - h_k e_j \in \mathcal{F}.$$

Proof. The points of the subsequence $\{x_k\}_K$ belong to the closed set $\mathcal{F}$, so that, $\bar{x} \in \mathcal{F}$. Since $\mathcal{F}$ is defined by box constraints, for $j = 1, \ldots n$ we have that at least one of the vectors e_j or $-e_j$ is a feasible direction at $\bar{x}$. Let $d_j \in \{e_j, -e_j\}$ be a feasible direction at $\bar{x}$, and let $D(\bar{x})$ be the set of feasible directions at $\bar{x}$. Again, since $\mathcal{F}$ is defined by linear constraints, from known results (see, e.g., Proposition 4 in [17] and Proposition A1 in [15]) it follows:

(i) $D(\bar{x}) \subseteq D(x_k)$ for $k \in K$ and k sufficiently large;

(ii) given $d \in D(\bar{x})$, there exists $\bar{t} > 0$ such that, for all $t \in (0, \bar{t}]$, we have $x_k + td \in \mathcal{F}$, for $k \in K$ and k sufficiently large.

Then for any $j = 1, \ldots, n$, being $d_j \in \{e_j, -e_j\}$ a feasible direction at $\bar{x}$ and recalling that $h_k \to 0$ for $k \in K$ and $k \to \infty$, it follows from (ii) that

$$x_k + h_k d_j \in \mathcal{F}$$

for $k \in K$ and k sufficiently large, and the thesis is proved. □

Finally, we can state the main theorem concerning the convergence of Algorithm `MOIF`.

Theorem 1 *Let $\{x_k\}$ be a sequence produced by Algorithm* `MOIF`*. Then $\{x_k\}$ admits limit points, and each one of them is Pareto stationary for Problem (1).*

Proof. By construction, $\{x_k\} \subset \mathcal{F}$. Hence, by compactness of $\mathcal{F}$ and using Lemma 3, there exists a limit point $\bar{x} \in \mathcal{F}$ of $\{x_k\}$ and an infinite subset of iteration indices $K \subset \{0, 1, 2 \ldots\}$ such that

$$\lim_{k \in K, k \to \infty} x_k = \bar{x}$$

and

$$\lim_{k \in K, k \to \infty} h_k = 0. \tag{13}$$

From Lemma 4 it follows that, for $k \in K$ and sufficiently large, procedure `imstepMulti`(x_k, h_k, τ, γ) can never return because $S(x_k, h_k) \cap \mathcal{F} = \emptyset$ or because $x_k + h_k e_i \notin \mathcal{F}$ and $x_k - h_k e_i \notin \mathcal{F}$, for all $i = 1, \ldots, n$, i.e. the stencil is feasible and all the components of the approximated gradient at x_k are finite as defined by (8).
Then, after relabelling (if necessary) the index set K, we can split K into two subsets, K_1 and K_2, defined as

$$\begin{aligned} K_1 &= \{k \in K : -\tau h_k \leq \theta(x_k, h_k) \leq 0\} \text{ and} \\ K_2 &= \{k \in K \setminus K_1 : \ \alpha_k |\theta(x_k, h_k)| \leq \tau h_k\}. \end{aligned}$$

Note that, since K is infinite, K_1 and K_2 cannot be both finite.
First, let us suppose that K_1 is infinite. From the definition of K_1, using assertion (ii) of Proposition 3 in the Appendix and (13), we obtain that

$$\lim_{k \to \infty, k \in K_1} \theta(x_k, h_k) = \theta(\bar{x}) = 0,$$

which completes the proof in this case.

Let us now suppose that K_2 is infinite. We proceed by contradiction and assume that $\bar{x}$ is not Pareto stationary, i.e.

$$\theta(\bar{x}) < 0. \tag{14}$$

Then, from the definition of K_2 and considering (13), we get

$$\lim_{k \in K_2, k \to \infty} \alpha_k |\theta(x_k, h_k)| = 0,$$

which, recalling (ii) of Proposition 3 in the Appendix, (13) and (14), yields

$$\lim_{k \in K_2, k \to \infty} \alpha_k = 0. \tag{15}$$

Furthermore, it must exist an infinite subset $\bar{K}_2 \subset K_2$ such that one of the following two cases can occur:

i) either $\alpha_k = 0$ for $k \in \bar{K}_2$

ii) or $\alpha_k = 2^{\beta_k} h_k > 0$ for $k \in \bar{K}_2$.

Point i). In this case, recalling that $v(x_k, h_k) = y(x_k, h_k) - x_k$ so that $x_k + v(x_k, h_k) = y(x_k, h_k) \in \mathcal{F}$ by definition, that for k sufficiently large $h_k < 1$, and that $\mathcal{F}$ is a convex set, we have that $x_k + h_k v_k \in \mathcal{F}$ for k sufficiently large and, by definition of the Goldstein procedure,

$$F(x_k + h_k v_k) \not\preceq F(x_k) + \mathbf{1}\gamma h_k \theta_k, \tag{16}$$

where we introduced the notations $v_k = v(x_k, h_k)$ and $\theta_k = \theta(x_k, h_k)$. If (16) holds, we can write, for at least an index $\ell \in \{1, \ldots, m\}$ (which can depend on k),

$$\frac{f_\ell(x_k) - f_\ell(x_k + h_k v_k)}{h_k} < -\gamma\theta_k \leq -\gamma \nabla_{h_k} f_\ell(x_k)^\top v_k.$$

Then, by the Mean Value Theorem, an $\tilde{h}_k \in (0, h_k)$ exists such that

$$-\nabla f_\ell(x_k + \tilde{h}_k v_k)^\top v_k < -\gamma\theta_k \leq -\gamma \nabla_{h_k} f_\ell(x_k)^\top v_k. \tag{17}$$

Moreover,

$$\begin{aligned} -\gamma \nabla_{h_k} f_\ell(x_k)^\top v_k &= \gamma(\nabla f_\ell(x_k) - \nabla_{h_k} f_\ell(x_k))^\top v_k - \gamma \nabla f_\ell(x_k)^\top v_k \\ &\leq \gamma \|\nabla f_\ell(x_k) - \nabla_{h_k} f_\ell(x_k)\| \|v_k\| - \gamma \nabla f_\ell(x_k)^\top v_k. \end{aligned}$$

Now, using (17) and the above relation, and recalling that the sequence of search directions $\{v_k\}$ is bounded, a fixed $\bar{\ell} \in \{1, \ldots, m\}$, a direction $\bar{v} \in \mathbb{R}^n$, and an infinite subset $K_3 \subseteq \bar{K}_2$ exist such that

$$-\nabla f_{\bar{\ell}}(x_k + \tilde{h}_k v_k)^\top v_k < -\gamma\theta_k \leq \gamma \|\nabla f_{\bar{\ell}}(x_k) - \nabla_{h_k} f_{\bar{\ell}}(x_k)\| \|v_k\| - \gamma \nabla f_{\bar{\ell}}(x_k)^\top v_k, \quad \forall k \in K_3, \tag{18}$$

and

$$\lim_{k \in K_3, k \to \infty} v_k = \bar{v}.$$

Taking the limit in (18) and recalling Proposition 2 and Proposition 3 in Appendix, we then obtain

$$-\nabla f_{\bar{\ell}}(\bar{x})^\top \bar{v} \ \leq -\gamma\theta(\bar{x}) \leq -\gamma \nabla f_{\bar{\ell}}(\bar{x})^\top \bar{v},$$

from which $(1-\gamma)\nabla f_{\bar{\ell}}(\bar{x})^\top \bar{v} \geq 0$ and $\theta(\bar{x}) \geq \nabla f_{\bar{\ell}}(\bar{x})^\top \bar{v}$ follow. Together with $\gamma \in (0, 1)$ these yield $0 > \theta(\bar{x}) \geq \nabla f_{\bar{\ell}}(\bar{x})^\top \bar{v} \geq 0$, i.e. a contradiction.

Point ii). In this case, $\alpha_k = 2^{\beta_k} h_k > 0$. From (15), recalling that v_k is computed in such a way that $x_k + v_k \in \mathcal{F}$, see e.g. (10), for $k \in \bar{K}_2$ and k sufficiently large we have $x_k + 2\alpha_k v_k \in \mathcal{F}$. Hence, the Goldstein procedure returns α_k such that, recalling (15) and the convexity of $\mathcal{F}$,

$$x_k + 2\alpha_k v_k \in \mathcal{F} \quad \text{and} \quad F(x_k + 2\alpha_k v_k) \not\preceq F(x_k) + \mathbf{1}\gamma(2\alpha_k \theta_k). \tag{19}$$

Then, reasoning as above, we can again conclude that $\bar{x}$ is Pareto stationary for Problem (1), thus raising yet a contradiction and concluding the proof. □

6 Computational experiments

In this section we report the numerical results of experiments performed in order to assess the effectiveness and the efficiency of the proposed algorithm, both when it is used to generate a single non-dominated point (i.e. when preferences of the decision maker are disregarded), and to generate a set of non-dominated solutions (i.e. when preferences of the decision maker are taken into account *a posteriori*). In both situations, our strategy is compared with the direct multisearch algorithm (DMS) proposed in [4]. We acknowledge that algorithm DFMO proposed in [16] is also suitable for the solution of Problem (1) when derivatives are not available. However, considering the results reported in [16] and in particular Figure 2 therein, DMS appears to be better than DFMO, at least when the coordinate directions are used as search directions in both algorithms.

Test problems. We considered the set of 100 multiobjective problems used in [4], whose dimension n is in the range $[1, 30]$ and with a number m of objectives belonging to the set $\{2, 3, 4\}$.

Implementation details. We have implemented Algorithm 4 in MATLAB© and we tested it on an Intel 1.2 GHz quad-core multithread with 8GB RAM. Parameters of algorithm `MOIF` have been set as follows

$$\begin{array}{ll} h_0 = 1, & \tau = 10^{-2} \\ \delta = 0.5, & \gamma = 10^{-5}. \end{array}$$

The computation of $\theta(x, h)$ and $y(x, h)$, as defined in (9) and (10), respectively, in Algorithm `MOIF` is carried out by using the `linprog` function of MATLAB©.
In the experiments, `MOIF` is stopped whenever either $h_k \leq 10^{-3}$ or the number of function evaluations exceeds 20,000. Furthermore, as concerns the implementation of algorithm `imstepMulti`, i.e. Algorithm 5, the instruction "Choose $y \in \mathcal{S}$" is realized so that y is the first point in $\mathcal{S}$ not dominated by any other point in $\mathcal{S}$.
As for DMS, all of its parameters (except for `Pareto_front`) have been set to their default values. In particular, we have

$$\begin{array}{ll} \texttt{stop\_alfa} = 1 & \texttt{tol\_stop} = 10^{-3} \\ \texttt{stop\_feval} = 1 & \texttt{max\_fevals} = 20,000 \end{array}$$

so that the stopping criteria of DMS are the same as those used in `MOIF`. Both solvers start from the centroid of the feasible region $\mathcal{F}$, i.e.

$$(x_0)_i = \frac{u_i + l_i}{2}, \quad \text{for all } i = 1, \ldots, n.$$

6.1 Computation of a single non-dominated solution

We compare the performance of our algorithm `MOIF` (i.e. Algorithm 4) with the one of DMS when it is used to generate a single non-dominated solution, i.e. when the calling parameter `Pareto_front` is set to 1.
The first thing that we observe is that, in 54 out of 100 problems, the Goldstein linesearch is never performed since the Pareto $h-$stationarity condition

$$\theta_j \geq -\tau h$$

is always satisfied. When this happens, the two algorithms `MOIF` and DMS show the same behaviour and produce the same non-dominated point.

In the remaining 46 problems, at least one linesearch is performed with success by algorithm `MOIF` during the optimization process. In 20 out of these 46 problems, `MOIF` determines a point that dominates the point computed by DMS, while the point provided by DMS never happens to

dominate the point determined by MOIF. Hence, we can conclude that these results show the good performance of MOIF in terms of quality of the computed solution.
In the 26 problems where the two solvers computed solutions that do not dominate each other, we have the following situation in terms of objective function evaluations. In 5 out of 26 test problems MOIF required a lower number of function evaluations than that required by DMS. However, this situation is not surprising since, as we may expect, the linesearch procedure of MOIF requires additional function evaluations with respect to the plain coordinate search performed by both solvers.
On the whole, we may conclude that algorithm MOIF, which combines a coordinate search phase with an implicit filtering strategy, shows a good ability to produce non-dominated solutions, confirming the viability of the proposed approach. It can also be noted that, due to the additional burden of the Goldstein linesearch, MOIF might be somewhat more expensive than DMS.

6.2 Computation of a set of non-dominated solutions

When an approximation of the Pareto front is required, like e.g. in *a posteriori* methods, drawing inspiration from DMS, we have defined a version of algorithm MOIF aimed at approximating the whole Pareto front which we call $\texttt{MOIF}_{front}$ (see Algorithm 7). In the following, we discuss the main aspects that distinguish $\texttt{MOIF}_{front}$ from MOIF (i.e. the proposed algorithm aimed at computing a single non-dominated solution) and DMS.
The main difference between MOIF and $\texttt{MOIF}_{front}$ is that every iteration of $\texttt{MOIF}_{front}$ is characterized by a set, say L_k, of point-stepsize pairs rather than the single pair (x_k, h_k), as in MOIF and as it is common in algorithms for single objective optimization.
Management of the sequence of sets $\{L_k\}$ is performed drawing inspiration both from DMS [4] and DFMO [16]. Since this aspect and, in particular, the generation of L_{k+1} starting from L_k is not trivial, we describe it with some detail.
For each k, let L_k be the following finite set

$$L_k = \{(x_i, h_i),\ x_i \in \mathcal{F}, h_i > 0, i = 1, \ldots, r_k\},$$

where $r_k = |L_k|$ and h_i is the (tentative) stepsize associated with point x_i. At each iteration k, a current pair $(x_i^k, h_i^k) \in L_k$ is selected (according to a certain criterion). Then, let

$$L'_k = \left((S(x_i^k, h_i^k) \cap \mathcal{F}) \times \{h_i^k\}\right) \cup L_k$$

and

$$\tilde{L}_k = \{(x_i, h_i) \in L'_k :\ \nexists (x_j, h_j) \in L'_k \text{ s.t. } F(x_j) \leq F(x_i)\}\,. \tag{20}$$

Then, the new set L_{k+1} is so defined:

$$L_{k+1} = \tilde{L}_k \qquad \text{when } \tilde{L}_k \neq L_k, \tag{21a}$$

$$L_{k+1} = \Big\{(x_i, h_i) \in L_k \cup \{(w_k, \alpha_k)\} : \nexists (x_j, h_j) \in L_k \text{ s.t. } F(x_j) \leq F(x_i)\Big\} \qquad \text{when } \tilde{L}_k = L_k, \theta_k < -\tau h_i^k \text{ and } \alpha_k|\theta_k| > \tau h_i^k \tag{21b}$$

$$L_{k+1} = L_k \setminus \{(x_i^k, h_i^k)\} \cup \{(x_i^k, \delta h_i^k)\} \qquad \text{otherwise} \tag{21c}$$

where $\theta_k = \theta(x_i^k, h_i^k)$, $v_k = v(x_i^k, h_i^k)$, $w_k = x_i^k + \alpha_k v_k$ and $\alpha_k = \texttt{Goldstein}(x_i^k, v_k, h_i^k, \theta_k, \gamma)$.
Note that when the algorithm is not able to find any new nondominated point, that is, when both the coordinate search does not produce any new nondominated point (i.e. $\tilde{L}_k = L_k$), and the Goldstein linesearch either is not performed (because either the approximated gradient is undetermined or $\theta_k \geq -\tau h_i^k$) or produces an unsuitable stepsize (i.e. $\alpha_k|\theta_k| \leq \tau h_i^k$), L_{k+1} is obtained from L_k by replacing the selected pair (x_i^k, h_i^k) by $(x_i^k, \delta h_i^k)$. The same is done in case of infeasibile stencil, that is when $L_k = L'_k = \tilde{L}_k$ because $S(x_i^k, h_i^k) \cap \mathcal{F} = \emptyset$.

Algorithm 7 $\texttt{MOIF}_{front}$

$\gamma, \delta, \tau \in (0,1)$, $L_0 = \{(x_i, h_i),\ x_i \in \mathcal{F}, h_i > 0, i = 1, \ldots, r_0\}$ initial set of nondominated points
for $k = 0, 1, \ldots$ **do**
  Select $(x_i^k, h_i^k) \in L_k$ and compute $\tilde{L}_k$ as in (20)
  if $\tilde{L}_k \neq L_k$ **then**
    Set $L_{k+1} = \tilde{L}_k$
  else
    Compute $\theta_k = \theta(x_i^k, h_i^k)$, $y_k = y(x_i^k, h_i^k)$, $v_k = y_k - x_i^k$
    if $\theta_k \geq -\tau h_i^k$ **then**
      Set $L_{k+1} = L_k \setminus \{(x_i^k, h_i^k)\} \cup \{(x_i^k, \delta h_i^k)\}$
    else
      Compute $\alpha_k = \text{Goldstein}(x_i^k, v_k, h_i^k, \theta_k, \gamma)$
      if $\alpha_k |\theta_k| \leq \tau h_i^k$ **then**
        Set $L_{k+1} = L_k \setminus \{(x_i^k, h_i^k)\} \cup \{(x_i^k, \delta h_i^k)\}$
      else
        Set L_{k+1} as in (21b)
      end if
    end if
  end if
end for

As we already pointed out in Remark 4, theoretical convergence properties of Algorithm $\texttt{MOIF}_{front}$ (at least under smoothness assumptions) would derive from the Goldstein linesearch procedure. However, carrying out such a theoretical convergence analysis for algorithm $\texttt{MOIF}_{front}$ would considerably burden the paper. Indeed, before proceeding with the theoretical analysis it would be necessary to formally state what a sequence of points is supposed to be in an algorithmic framework that generates sequences of sets of points. Most probably, it could be necessary to proceed as in [16] where "linked sequences of points" are defined. Then, stationarity results could be given with reference to (some) linked sequences.

Numerical results. Comparison of $\texttt{MOIF}_{front}$ and DMS in this context is carried out by means of the Purity and Spread (both Γ and Δ) metrics used in [4], and by using performance profiles [6]. We recall that the purity metric measures the quality of the generated front, i.e. how good the non-dominated points computed by a solver are with respect to those computed by any other solver. Note that, for each problem p, the "reference" Pareto front F_p is calculated by first computing

$$F'_p = F_{p,DMS} \cup F_{p,\texttt{MOIF}},$$

where $F_{p,s}$ denotes the set of non-dominated solutions found by solver s, and then removing from this set any dominated solution, that is

$$F_p = \{\, x \in F'_p : \nexists y \in F'_p \ s.t.\ F(y) \leq F(x) \,\}. \tag{22}$$

On the other hand, the spread metrics are essential to measure the uniformity of the generated front in the objectives space.
In the experiments $\texttt{MOIF}_{front}$ is stopped at iteration k, when the number of function evaluations exceeds 20,000, or when the following stepsize criterion holds:

$$\max_{(x_i,h_i)\in L_k} h_i \leq 10^{-3}. \tag{23}$$

As concerns the choice of $x_k \in L_k$, we select the point with the highest Γ value as in [4].
In figure 1, comparison between $\texttt{MOIF}_{front}$ and DMS is reported in terms of the above mentioned metrics.

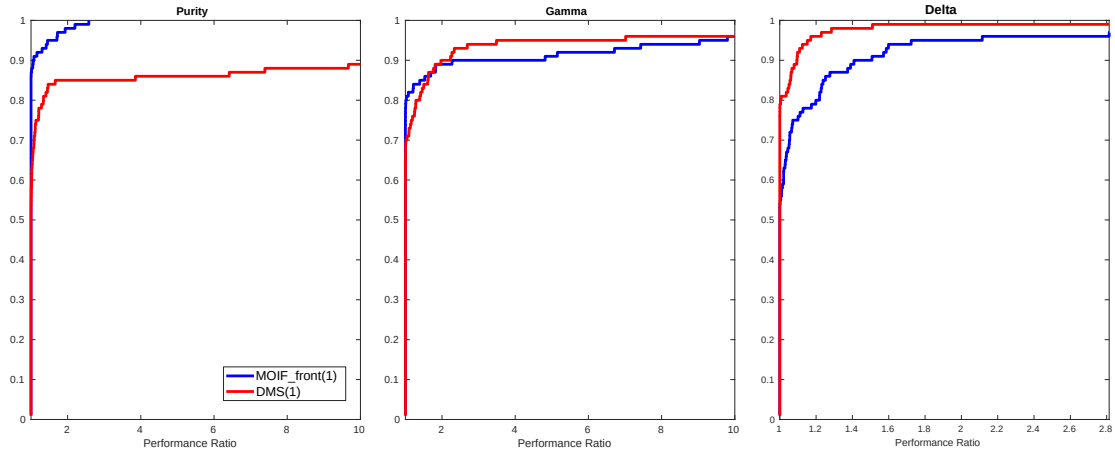

Figure 1: Comparison between $\texttt{MOIF}_{front}$ and DMS when both solvers start from the centroid of the feasible region $\mathcal{F}$

As we can see, $\texttt{MOIF}_{front}$ outperforms DMS in terms of purity, while it can be considered equivalent in terms of spread Γ. However, DMS outperforms $\texttt{MOIF}_{front}$ in terms of spread Δ.
As we may expect, the linesearch phase is very effective when quality of the generated points is regarded. This is confirmed by the results in terms of purity. We observe that many points of the current list L_k may be dominated by the points generated by the linesearch along "good" descent directions. Hence, the cardinality of the list of nondominated points may tend to quickly reduce with respect to a strategy based on the pure coordinate search as in DMS. This may lead to a generated front with a lower degree of uniformity compared with that generated by DMS.
In order to take into account the effect of the linesearch in terms of the spread metrics, we defined a version of the algorithm that uses a suitable condition to decide whether or not to perform the linesearch at a given iteration. More specifically, let $c_p \in [0, 1]$ be a parameter which we call purity coefficient. Then, in Algorithm $\texttt{MOIF}_{front}$, given the current pair $(x_i^k, h_i^k) \in L_k$ and θ_k, the Goldstein linesearch is performed only if

$$\theta_k < -\bar{\tau} h_i^k \quad \text{where} \quad \bar{\tau} = \begin{cases} \tau & \text{if } h_i^k \leq c_p \max_{(x,h)\in L_k}\{h\}, \\ \infty & \text{otherwise.} \end{cases}$$

Note that, setting $\bar{\tau} = \infty$ will make the test before the Goldstein linesearch in Algorithm $\texttt{MOIF}_{front}$ surely satisfied so that the Goldstein linesearch is not executed. Furthermore, it is worth noting that, when $c_p = 1$, then the Goldstein linesearch is performed just as in the original version of $\texttt{MOIF}_{front}$, while if $c_p = 0$ then, as in DMS, no linesearch is ever executed.
The idea underlying the use of the above condition is that of managing the uniformity of the generated front by controlling the uniformity of the sampling step sizes related to the points of the list. As we can see from the performance profiles reported in figures 2 and 3, when we reduce c_p our algorithm became closer to DMS solver.
Finally, we observe that, when $c_p = 0$, Algorithm $\texttt{MOIF}_{front}$ becomes equal to a specific instance of DMS, namely the one using the set of directions $\{e_1, \ldots, e_n, -e_1, \ldots, -e_n\}$.

6.3 Comparison using a set of initial points

In this subsection, we compare $\texttt{MOIF}_{front}$ and DMS choosing a set of initial solutions rather than a single point. In particular, rather than starting the solvers from the single point x_0 as we did in subsections 6.1 and 6.2, we let them start from a list of n points equally spaced on the line segment

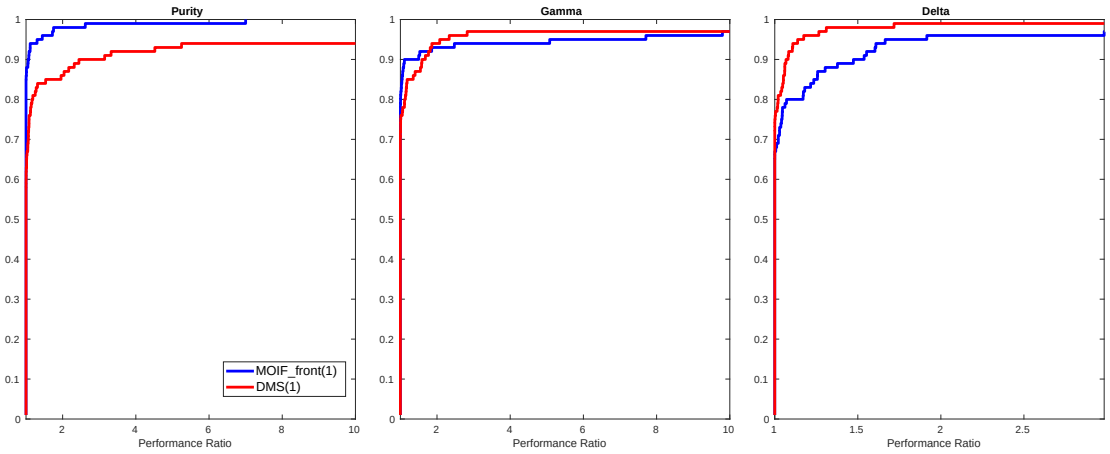

Figure 2: Comparison between $\texttt{MOIF}_{front}$, with $c_p = 0.5$, and DMS

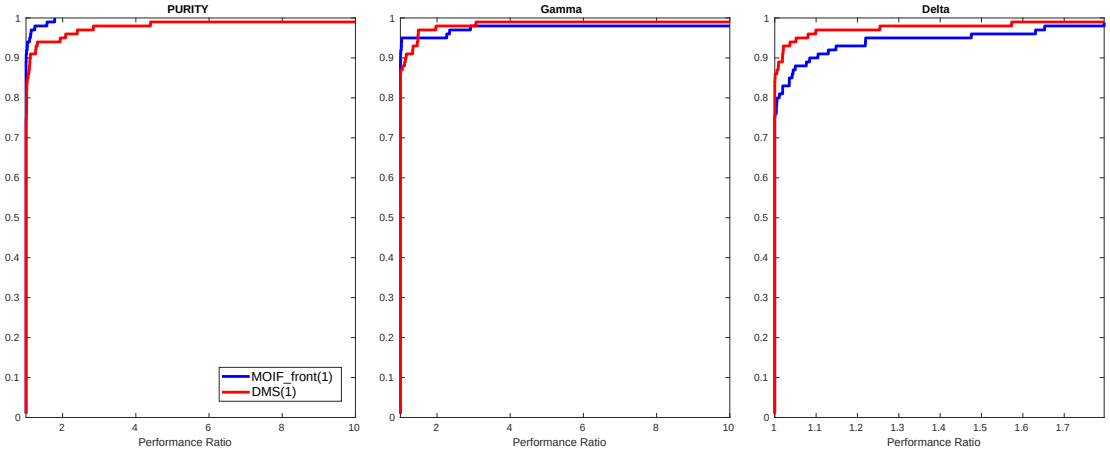

Figure 3: Comparison between $\texttt{MOIF}_{front}$, with $c_p = 0.2$, and DMS

joining the upper and lower bounds on the variables. Specifically, we let

$$\tilde{L}_0 = \{(x_0^1, h_0), \ldots, (x_0^n, h_0)\}$$

with

$$(x_0^j)_i = l_i + \frac{u_i - l_i}{n-1}(j-1), \quad \text{for} \quad i, j = 1, \ldots, n.$$

Then,

$$L_0 = \{(x_i, h_i) \in \tilde{L}_0 : \nexists\, (x_j, h_j) \in \tilde{L}_0 \text{ s.t. } F(x_j) \leq F(x_i)\},$$

i.e. the set obtained from $\tilde{L}_0$ by removing dominated solutions. Note that, for DMS, this is version DMS(n,line), as defined in [4].

The performance profiles are reported in figure 4. As in the previous case, we can still say that $\texttt{MOIF}_{front}$ outperforms DMS in terms of purity. As for the spread metrics, the two solvers are almost equivalent in terms of spread Γ, while $\texttt{MOIF}_{front}$ is outperformed by DMS in terms of spread Δ. However, it is worth noting that the superiority of $\texttt{MOIF}_{front}$ over DMS in terms of

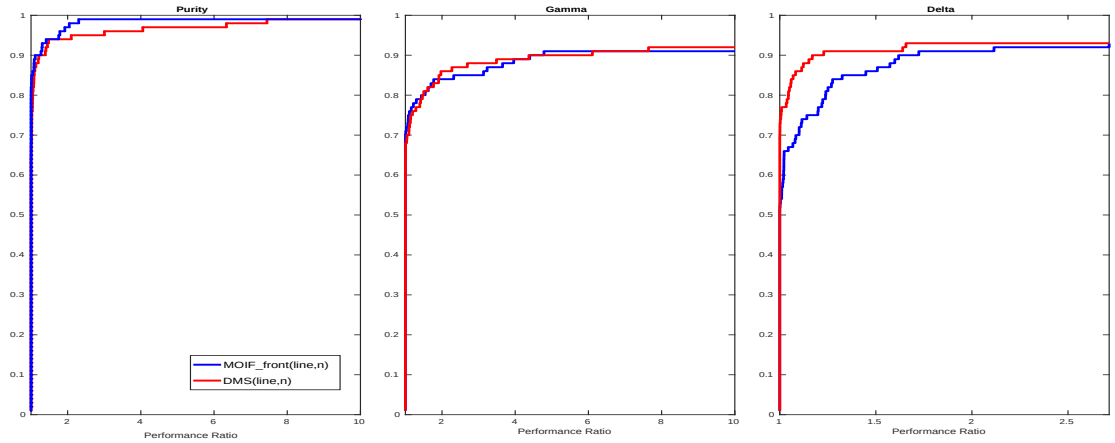

Figure 4: Comparison between $\texttt{MOIF}_{front}$, with $c_p = 1$, and DMS when both solvers start from set L_0

purity is considerably reduced with respect to the single starting point case (see figure 1).
This situation brings us to argue that the use of the initial set L_0 seems to help DMS to generate better estimates of the Pareto front. This is indeed the case with DMS starting from set L_0 beating DMS starting from the centroid in terms of purity but at the expense of the spread metrics, for which the version of DMS starting from the centroid seems the best one. On the contrary, $\texttt{MOIF}_{front}$ seems not to be able to produce better estimates when it starts from the set L_0 rather than from the single point x_0.This can in turn be explained by recalling the 20,000 function evaluations limit. Indeed, at least for problems where the implicit filtering stepsize h_k is bigger than the tolerance 10^{-3} when the above limit is hit, the final estimate of the Pareto front could still be far away from the "real" one. This is to say that the linesearches employed by MOIF have the effect to consume more function evaluations with respect to the "simpler" pattern search strategy of DMS.
Finally, we have considered the comparison between the best version of $\texttt{MOIF}_{front}$, i.e. the one with $c_p = 1$ and starting from the centroid of the feasible region (as evidenced in Figure 5) and the default version of DMS, i.e. the one starting from set L_0. Performance profiles for the purity and spread metrics relative to this further comparison are reported in Figure 6. From the figure, it is apparent the superiority of $\texttt{MOIF}_{front}$ (starting from the centroid) over DMS (starting from set L_0) both in terms of purity and spread Γ. As concerns the spread Δ, it emerges quite a new situation with respect to the already seen comparisons. In fact, spread Δ profiles reported in Figure 6 show

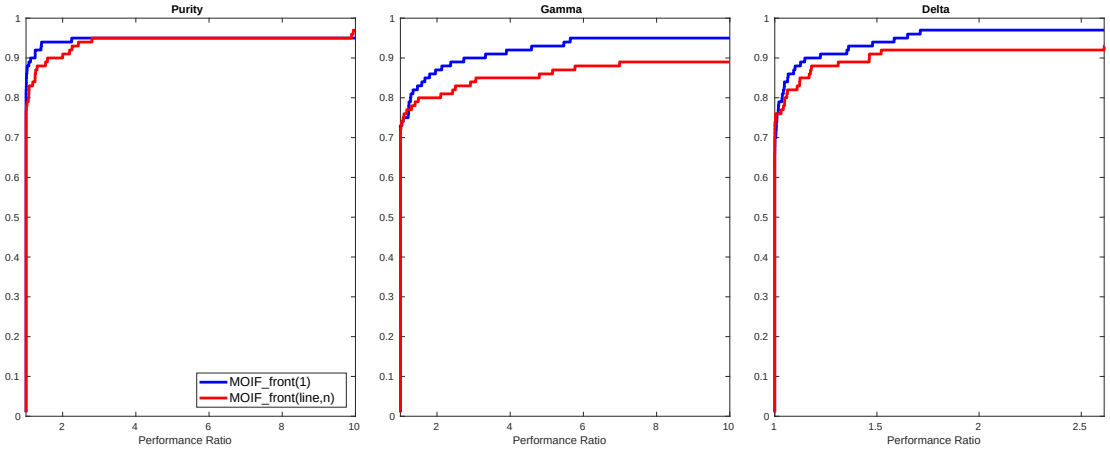

Figure 5: Comparison between $\texttt{MOIF}_{front}$ with $c_p = 1$, starting from the centroid of $\mathcal{F}$ and from set L_0

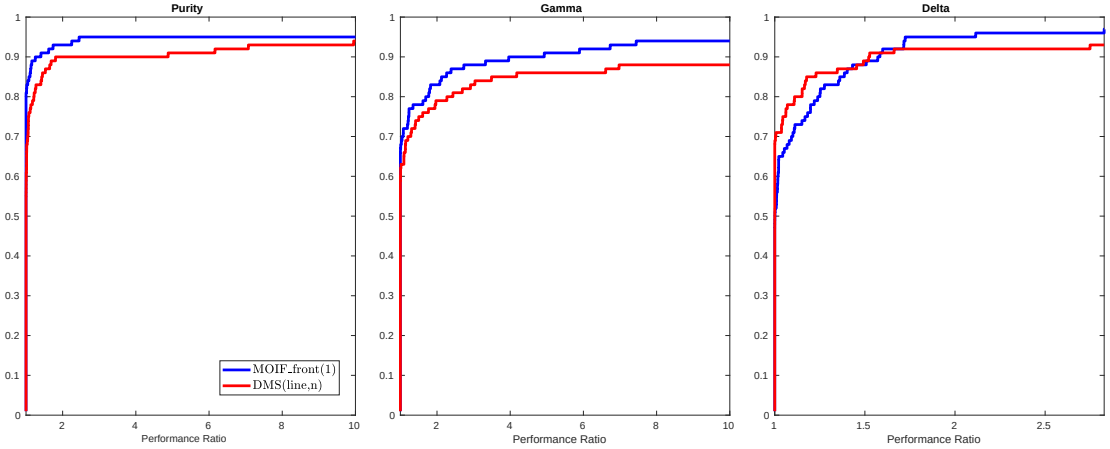

Figure 6: Comparison between $\texttt{MOIF}_{front}$, with $c_p = 1$ and starting from the centroid of $\mathcal{F}$, and DMS starting from set L_0

that DMS is clearly less robust than $\texttt{MOIF}_{front}$ even though the former method is more efficient than the latter one.

6.4 Numerical results with noisy functions

In this section we report the numerical results obtained by $\texttt{MOIF}_{front}$ in the case of noisy functions. In our experiments we consider additive noise sampled from a normal distribution. For every feasible point x, we evaluate a noisy version $\tilde{F} : \mathbb{R}^n \to \mathbb{R}^m$ of F such that

$$\tilde{F}(x) := F(x) + \mathcal{N}(0, \varepsilon\bar{\sigma}^2), \tag{24}$$

where $\varepsilon \in \mathbb{R}^+$ is a smoothing parameter which controls the noise level, $\bar{\sigma}^2 \in \mathbb{R}^m$ is such that

$$\bar{\sigma}^2 := \begin{bmatrix} |f_1^{max}| \\ \vdots \\ |f_m^{max}| \end{bmatrix}, \tag{25}$$

and, for each problem p, f_i^{max} is the maximum value of the $i-$th objective on the reference Pareto front F_p defined in (22).
In order to give a good estimation of the gap between the Pareto front found in the noisy case and the reference Pareto front, we consider the *Generational Distance* (GD) metric introduced in [19]:

$$GD = \frac{\left(\sum_{i=1}^{P} d_i^2\right)^{1/2}}{P}, \tag{26}$$

where P is the number of points found by a solver, $x_1, x_2, \ldots, x_P$, and d_i is the euclidean distance of the corresponding noise-free value $F(x_i)$ from the *real* Pareto front. GD is a recommended metric in test problems for which a set of Pareto optimal solutions is known. Although in our test problems the real Pareto front is generally unknown, we use it in order to show the extent of convergence of $\texttt{MOIF}_{front}$ and DMS with respect to their noise-free versions. Therefore we adopted in (26) the following definition of d_i:

$$d_i = \begin{cases} 0 & \text{if } x_i \text{ is non-dominated by } F_p \\ \min\limits_{x \in F_p} \{||F(x) - F(x_i)||_2\} & \text{otherwise.} \end{cases} \tag{27}$$

We remark that the contribution of d_i to the GD metric is zero not only if x_i belongs to the reference Pareto front, but also if it is better (this may happen because F_p is an approximation of the real Pareto Front). We also remark that GD, as the Purity metric, does not consider the quantity of generated points, but only the quality.
We tested both $\texttt{MOIF}_{front}$ and DMS, starting from the centroid of the feasible region $\mathcal{F}$ using the stopping criteria defined in section 6.2 and $\varepsilon \in \{0.0001, 0.05, 0.1, 0.2\}$.
We did not observe significant differences in the performance of the two algorithms for $\varepsilon \in \{0.05, 0.1, 0.2\}$. Therefore, we report the results only for $\varepsilon = 0.0001$. Figure 7 shows the good performance of $\texttt{MOIF}_{front}$ in terms of GD metric compared with those of DMS. For the same value of $\varepsilon = 0.0001$ we have also compared $\texttt{MOIF}_{front}$ and DMS in terms of purity and spread metrics over the 1000 instances. The results of the comparison are reported in figure 8. We may observe that the results of the comparison are similar to those obtained without noise and reported in figure 1, that is, $\texttt{MOIF}_{front}$ outperforms DMS in terms of the purity metric, while DMS outperforms $\texttt{MOIF}_{front}$ in terms of spread metrics. On the whole, $\texttt{MOIF}_{front}$ and DMS show a similar robustness in presence of moderate noise.

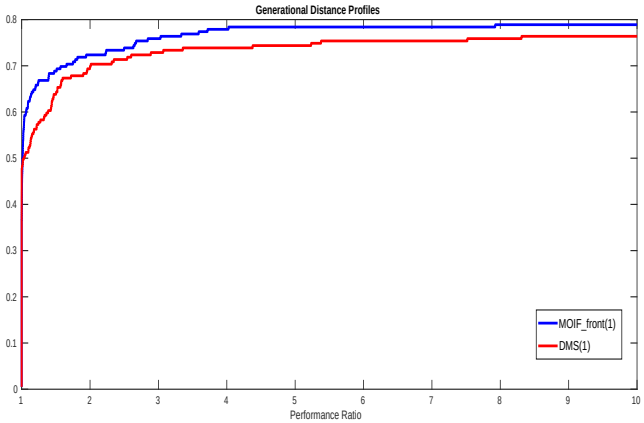

Figure 7: Generational Distance comparison between $\texttt{MOIF}_{front}$ and DMS, starting from the centroid, with $\varepsilon = 0.0001$

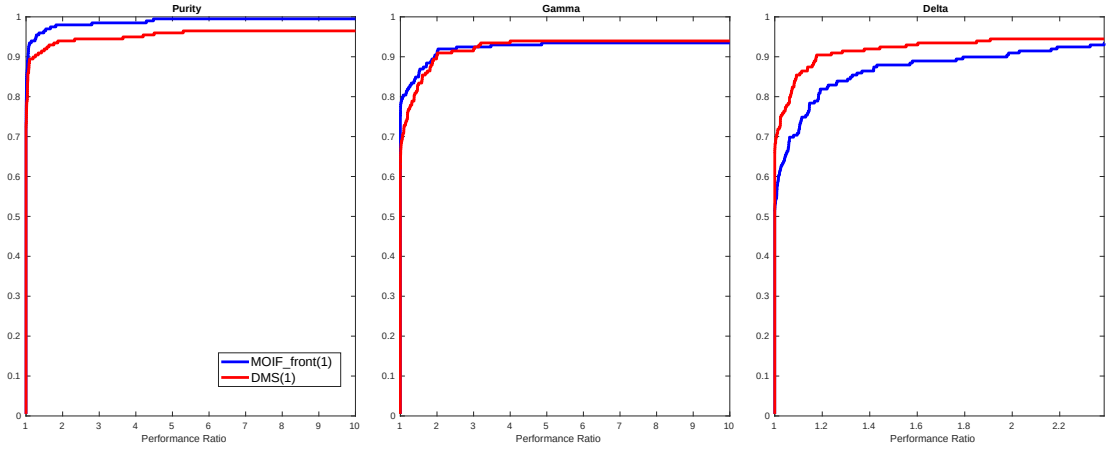

Figure 8: Purity and spread metrics comparison between $\texttt{MOIF}_{front}$ and DMS, starting from the centroid, with $\varepsilon = 0.0001$

7 Concluding remarks

In this work a derivative-free algorithm for smooth multiobjective optimization has been proposed. The algorithm combines a coordinate search with a suitable extension of the implicit filtering strategy to multiobjective optimization with box constraints. Global convergence results are established under standard assumptions. The results of the computational experiments and the comparison with a state-of-art algorithm show the effectiveness of the proposed algorithm both in computing a single Pareto solution and in reconstructing the entire Pareto front. The approach can be easily and advantageously adapted to the case of multiobjective optimization problems where the derivatives of some objective functions are available.
The case when some (possibly all) of the objective functions are only Lipschitz continuous in the feasible set is the subject of future work. In the following we briefly discuss the main differences between the smooth and the nonsmooth case.
In the smooth case, thanks to the approximation of the gradient by finite-differences, the implicit filtering phase generates a search direction, by solving the min max problem (see (10)), approximating the steepest direction and this, coupled by employment of the line search, allowed us to attain convergence properties. In the nonsmooth case, in order to guarantee theoretical properties, we need to make use of a (pre-determined) set of directions which is dense in the unit sphere. We may expect that the implicit filtering phase could be suitably adapted to evaluate the "goodness" of a given search direction to decide whether or not to perform a line search along it. Then, the extension of the proposed approach to the nonsmooth case seems to lead to a substantially different algorithmic framework, and therefore it is the subject of ongoing research and the topic of a future paper.

Acknowledgements

We are thankful to three anonymous reviewers whose stimulating comments and suggestions greatly helped us improving the paper. Also, we would like to thank Prof. Ana Luísa Custódio, José F. Aguilar Madeira, A. Ismael F. Vaz, and Luís Nunes Vicente for providing us the matlab code of their direct multisearch algorithm (DMS). Work partially supported by INDAM-GNCS.

References

[1] K.R. Bailey and B.G. Fitzpatrick. Estimation of groundwater flow parameters using least squares. *Mathematical and Computer Modelling*, 26(11):117–127, 1997.

[2] R.G. Carter, J.M. Gablonsky, A. Patrick, C.T. Kelley, and O.J. Eslinger. Algorithms for noisy problems in gas transmission pipeline optimization. *Optimization and engineering*, 2(2):139–157, 2001.

[3] A.R. Conn, K. Scheinberg, and L.N. Vicente. *Introduction to derivative-free optimization*. Society for Industrial and Applied Mathematics, 2009.

[4] A.L. Custódio, J.F.A. Madeira, A.I.F. Vaz, and L.N. Vicente. Direct multisearch for multiobjective optimization. *SIAM Journal on Optimization*, 21(3):1109–1140, 2011.

[5] J. David, R.L. Ives, H.T. Tran, T. Bui, and M.E. Read. Computer optimized design of electron guns. *IEEE Transactions on Plasma Science*, 36(1):156–168, 2008.

[6] E.D. Dolan and J.J. Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213, 2002.

[7] L. dos Santos Coelho and V.C. Mariani. Combining of differential evolution and implicit filtering algorithm applied to electromagnetic design optimization. In *Soft Computing in Industrial Applications*, pages 233–240. Springer, 2007.

[8] J. Fliege and B.F. Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.

[9] K.R. Fowler, C.T. Kelley, C.E. Kees, and C.T. Miller. A hydraulic capture application for optimal remediation design. *Developments in Water Science*, 55:1149–1157, 2004.

[10] K.R. Fowler, C.T. Kelley, C.T. Miller, C.E. Kees, R.W. Darwin, J.P. Reese, M.W. Farthing, and M.S.C. Reed. Solution of a well-field design problem with implicit filtering. *Optimization and Engineering*, 5(2):207–234, 2004.

[11] M. Gen, R. Cheng, and L. Lin. Multiobjective genetic algorithms. In *Network Models and Optimization: Multiobjective Genetic Algorithm Approach*, pages 1–47. Springer, 2008.

[12] P. Gilmore and C.T. Kelley. An implicit filtering algorithm for optimization of functions with many local minima. *SIAM Journal on Optimization*, 5(2):269–285, 1995.

[13] P. Gilmore, C.T. Kelley, C.T. Miller, and G.A. Williams. Implicit filtering and optimal design problems. In *Optimal Design and Control*, pages 159–176. Springer, 1995.

[14] C.T. Kelley. *Implicit Filtering*. Society for Industrial and Applied Mathematics, 2011.

[15] C.-J. Lin, S. Lucidi, L. Palagi, A. Risi, and M. Sciandrone. Decomposition algorithm model for singly linearly constrained problems subject to lower and upper bounds. *Journal of Optimization Theory and Applications*, 141:107–126, 2009.

[16] G. Liuzzi, S. Lucidi, and F. Rinaldi. A derivative-free approach to constrained multiobjective nonsmooth optimization. *SIAM Journal on Optimization*, 26(4):2744–2774, 2016.

[17] S. Lucidi, L. Palagi, A. Risi, and M. Sciandrone. A convergent decomposition algorithm for support vector machines. *Computational Optimization and Applications*, 38:217–234, 2007.

[18] K. Miettinen. *Nonlinear Multiobjective Optimization*. International Series in Operations Research & Management Science. Springer, 1998.

[19] David Allen Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Wright Patterson AFB, OH, USA, 1999. AAI9928483.

[20] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P.N. Suganthan, and Q. Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.

8 Appendix: technical results

In the appendix we prove two technical results that are used for the convergence analysis.

Proposition 2 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable and let $x \in \mathcal{F}$. Let $\{z_k\} \subset \mathcal{F}$ and $\{h_k\} \subset \mathbb{R}^+$ be sequences such that*

$$\lim_{k\to\infty} z_k = x \qquad \lim_{k\to\infty} h_k = 0. \tag{28}$$

Assume that, for $i = 1, \ldots, n$, at least one of the following condition holds

$$z_k + h_k e_i \in \mathcal{F},$$

$$z_k - h_k e_i \in \mathcal{F}.$$

Then we have

$$\lim_{k\to\infty} \nabla_{h_k} f(z_k) = \nabla f(x).$$

Proof. Let $i \in \{1, \ldots, n\}$ and define the following subsets

$$\begin{aligned} K_1 &= \{k : z_k + h_k e_i \in \mathcal{F},\ z_k - h_k e_i \notin \mathcal{F}\}, \\ K_2 &= \{k : z_k \pm h_k e_i \in \mathcal{F}\}, \\ K_3 &= \{k : z_k - h_k e_i \in \mathcal{F},\ z_k + h_k e_i \notin \mathcal{F}\}. \end{aligned}$$

By definition of approximated gradient we have

$$\frac{\partial_h f(z_k)}{\partial x_i} = \begin{cases} \frac{f(z_k+h_k e_i)-f(z_k)}{h_k} & k \in K_1 \\ \frac{f(z_k+h_k e_i)-f(z_k-h_k e_i)}{2h_k} & k \in K_2 \\ \frac{f(z_k)-f(z_k-h_k e_i)}{h_k} & k \in K_3 \end{cases}$$

Suppose that K_1 is an infinite subset. For all $k \in K_1$, by the Mean Value Theorem, we can write

$$\frac{\partial_h f(z_k)}{\partial x_i} = \frac{\partial f(\xi_k)}{\partial x_i},$$

where $\xi_k = z_k + \theta_k h_k e_i$, with $\theta_k \in (0, 1)$. Taking the limits for $k \in K_1$ and $k \to \infty$, recalling (28) and the continuity of the gradient, we obtain

$$\lim_{k\in K_1, k\to\infty} \frac{\partial_h f(z_k)}{\partial x_i} = \frac{\partial f(x)}{\partial x_i}.$$

By repeating the same reasonings using the sets K_2 and K_3, we have

$$\lim_{k\to\infty} \frac{\partial_h f(z_k)}{\partial x_i} = \frac{\partial f(x)}{\partial x_i},$$

and the thesis is proved. □

Proposition 3 *Consider Problem* (1)*, let* $F : \mathbb{R}^n \to \mathbb{R}^m$ *be continuously differentiable,* $x \in \mathcal{F}$*, and let* $\theta : \mathcal{F} \times R^+ \to \mathbb{R}$ *be defined as in* (9)*. Then:*

(i) $\theta(x, h) \leq 0$ *for all* $x \in \mathcal{F}$ *and* $h > 0$*;*

(ii) *let* $\{z_k\} \subset \mathcal{F}$ *and* $\{h_k\} \subset \mathbb{R}^+$ *be sequences satisfying the assumptions of Proposition* 2*; we have*

$$\lim_{k\to\infty} \theta(z_k, h_k) = \theta(x).$$

Proof. (i) Given $x, y \in \mathcal{F}$ and $h > 0$, we consider the function g defined as follows:

$$g(y, h, x) = \max_{i=1,\ldots,m} \nabla_h f_i(x)^\top (y - x),$$

and note that

$$\theta(x, h) = \min_{y\in\mathcal{F}} g(y, h, x).$$

Then $\theta(x, h) \leq 0$ follows easily from $g(x, h, x) = 0$.

(ii) We preliminary observe that

$$|\max_i a_i - \max_i b_i| \leq \|a - b\|, \quad \text{for any } a, b \in \mathbb{R}^m.$$

Let us define

$$y(x) \in \arg\min_{y\in\mathcal{F}} \max_{i=1,\dots,m} \nabla f_i(x)^\top (y-x),$$

$$y_k \in \arg\min_{y\in\mathcal{F}} \max_{i=1,\dots,m} \nabla_{h_k} f_i(z_k)^\top (y-z_k),$$

so that

$$\max_{i=1,\dots,m} \nabla f_i(x)^\top (y(x)-x) \le \max_{i=1,\dots,m} \nabla f_i(x)^\top (y_k-x)$$

$$\max_{i=1,\dots,m} \nabla_{h_k} f_i(z_k)^\top (y_k-z_k) \le \max_{i=1,\dots,m} \nabla_{h_k} f_i(z_k)^\top (y(x)-z_k).$$

Denote by $J_{h_k}(z_k)$ the approximated Jacobian $J_{h_k}(z_k) = [\nabla_{h_k} f_1(z_k), \dots, \nabla_{h_k} f_m(z_k)]^\top$. We can write

$$\begin{aligned}
\theta(z_k,h_k)-\theta(x) &= \max_i \nabla_{h_k} f_i(z_k)^\top (y_k-z_k) - \max_i \nabla f_i(x)^\top (y(x)-x) \\
&\le \max_i \nabla_{h_k} f_i(z_k)^\top (y(x)-z_k) - \max_i \nabla f_i(x)^\top (y(x)-x) \\
&\le \|J_{h_k}(z_k)^\top (y(x)-z_k) - J(x)^\top (y(x)-x)\| \\
&\le \|(J_{h_k}(z_k)-J(x))^\top y(x)\| + \|J(x)^\top x - J_{h_k}(z_k)^\top z_k + J(x)^\top z_k - J(x)^\top z_k\| \\
&\le \|(J_{h_k}(z_k)-J(x))^\top y(x)\| + \|J(x)^\top (z_k-x)\| + \|(J_{h_k}(z_k)-J(x))^\top z_k\|.
\end{aligned}$$

A quite similar bound, with y_k in place of $y(x)$, can be obtained for $\theta(x)-\theta(z_k,h_k)$. Then, as z_k and y_k belong to the compact set $\mathcal{F}$, by Proposition 2, $|\theta(z_k,h_k)-\theta(x)| \to 0$ for $k\to\infty$. □